### Кам’янець-Подільський національний університет імені Івана Огієнка

### КАФЕДРА КОМП’ЮТЕРНИХ НАУК

### Навчальна дисципліна «Об’єктно-орієнтоване програмування»

### ЛАБОРАТОРНА РОБОТА #0108

## Тема:

Робота з файлами

Варіант №2

**Виконав**:  
студент 1-го курсу  
групи KNms1-B23  
Веретко С. І.

**Прийняла**:  
старший викладач,  
Мястковська М.О.

Кам’янець-Подільський – 2023

1. Теоретичні відомості

Більшість задач у програмуванні так чи інакше пов’язані з роботою з файлами та каталогами. Нам може знадобитися прочитати текст із файлу або отримати запис, видалити файл чи весь каталог, не кажучи вже про більш комплексні завдання, як, наприклад, створення текстового редактора та інших подібних завдань.

# Фреймворк .NET надає великі можливості для управління та маніпуляцій файлами та каталогами, які є більшими частинами сосредоточени в просторі імені System.IO . Класи, розташовані в цьому просторі (такі як Stream, StreamWriter, FileStream тощо), дозволяють керувати файлами вводом-виводом.

Для представлення диску в просторі імен : **System.IO** є клас **DriveInfo** .

Цей клас має статичний метод GetDrives, який повертає імена всіх логічних дисків комп'ютера. Також він пропонує ряд корисних властивостей:

* AvailableFreeSpace : вказує на обсяг доступного вільного місця на диску в байтах
* DriveFormat : отримує ім'я файлової системи
* DriveType : представляє тип диска
* IsReady : готовий диск (наприклад, DVD-диск може бути не вставлений в дисковод)
* Ім'я : отримує ім'я диска
* TotalFreeSpace : отримує загальний обсяг вільного місця на диску в байтах
* TotalSize : загальний розмір диска в байтах
* VolumeLabel : отримує або встановлює метку тома

2. Завдання для виконання

### Завдання 1.

Реалізувати наступні операції з файловою системою:

1) На диску D: створіть каталог OOP\_lab08.

2) Cтворіть у ньому наступні каталоги (назви англійською мовою):

* Номер\_вашої\_групи (підставте свою групу);
* Ваше\_прізвище (підставте своє прізвище);
* Sources;
* Reports;
* Texts.

3) Скопіюйте каталоги Texts, Sources та Reports до каталогу Ваше\_прізвище.

4) Перемістіть каталог Ваше\_прізвище до каталогу Номер\_вашої\_групи.

5) У каталозі Texts створіть текстовий файл dirinfo.txt у який помістіть усю інформацію про даний каталог (див. приклад вище).

### Завдання 2.

Дано файл f, компоненти якого є дійсними числами. Знайти:

1) найбільше із значень компонент;

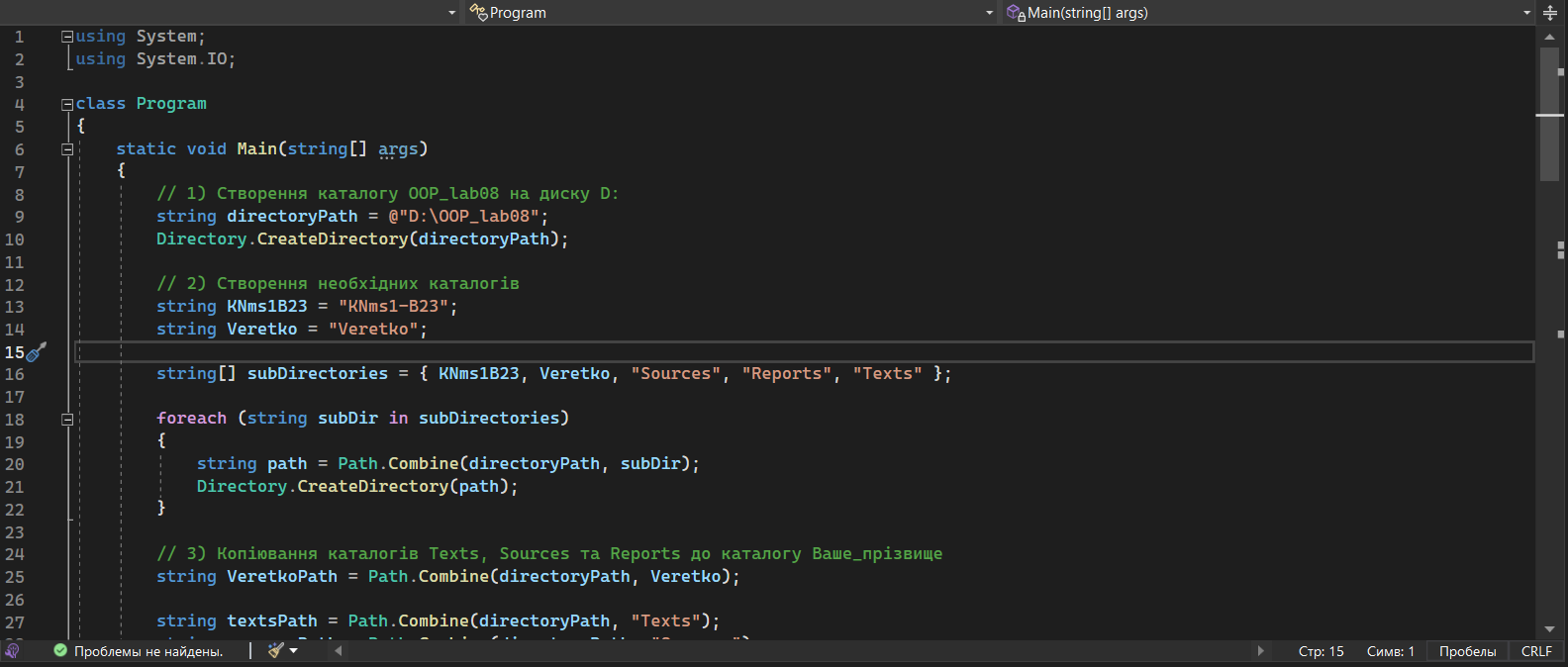
2) найменше із значень компонент з парними номерами;

3) найбільше із значень модулів компонент з непарними номерами;

4) суму найбільшого і найменшого із значень компонент;

5) різницю першої і останньої компонент файлу.

3. Послідовність виконання завданнь



4. Лістинг програми

Завдання 8\_1

using System;

using System.IO;

class Program

{

static void Main(string[] args)

{

// 1) Створення каталогу OOP\_lab08 на диску D:

string directoryPath = @"D:\OOP\_lab08";

Directory.CreateDirectory(directoryPath);

// 2) Створення необхідних каталогів

string KNms1B23 = "KNms1-B23";

string Veretko = "Veretko";

string[] subDirectories = { KNms1B23, Veretko, "Sources", "Reports", "Texts" };

foreach (string subDir in subDirectories)

{

string path = Path.Combine(directoryPath, subDir);

Directory.CreateDirectory(path);

}

// 3) Копіювання каталогів Texts, Sources та Reports до каталогу Ваше\_прізвище

string VeretkoPath = Path.Combine(directoryPath, Veretko);

string textsPath = Path.Combine(directoryPath, "Texts");

string sourcesPath = Path.Combine(directoryPath, "Sources");

string reportsPath = Path.Combine(directoryPath, "Reports");

DirectoryCopy(textsPath, Path.Combine(VeretkoPath, "Texts"), true);

DirectoryCopy(sourcesPath, Path.Combine(VeretkoPath, "Sources"), true);

DirectoryCopy(reportsPath, Path.Combine(VeretkoPath, "Reports"), true);

// 4) Переміщення каталогу Ваше\_прізвище до каталогу Номер\_вашої\_групи

string groupPath = Path.Combine(directoryPath, KNms1B23);

Directory.Move(VeretkoPath, Path.Combine(groupPath, Veretko));

// 5) Створення текстового файлу dirinfo.txt у каталозі Texts з інформацією про каталог Texts

string dirInfoFilePath = Path.Combine(directoryPath, "Texts", "dirinfo.txt");

DirectoryInfo textsDirectoryInfo = new DirectoryInfo(textsPath);

using (StreamWriter writer = new StreamWriter(dirInfoFilePath))

{

writer.WriteLine($"Full path: {textsDirectoryInfo.FullName}");

writer.WriteLine($"Creation time: {textsDirectoryInfo.CreationTime}");

writer.WriteLine($"Last access time: {textsDirectoryInfo.LastAccessTime}");

writer.WriteLine($"Number of files: {textsDirectoryInfo.GetFiles().Length}");

writer.WriteLine($"Number of subdirectories: {textsDirectoryInfo.GetDirectories().Length}");

}

Console.WriteLine("Операції з файловою системою виконані успішно!");

}

// Метод для рекурсивного копіювання вмісту каталогу

static void DirectoryCopy(string sourceDirName, string destDirName, bool copySubDirs)

{

DirectoryInfo dir = new DirectoryInfo(sourceDirName);

DirectoryInfo[] dirs = dir.GetDirectories();

if (!dir.Exists)

{

throw new DirectoryNotFoundException($"Source directory does not exist or could not be found: {sourceDirName}");

}

if (!Directory.Exists(destDirName))

{

Directory.CreateDirectory(destDirName);

}

FileInfo[] files = dir.GetFiles();

foreach (FileInfo file in files)

{

string temppath = Path.Combine(destDirName, file.Name);

file.CopyTo(temppath, false);

}

if (copySubDirs)

{

foreach (DirectoryInfo subdir in dirs)

{

string temppath = Path.Combine(destDirName, subdir.Name);

DirectoryCopy(subdir.FullName, temppath, copySubDirs);

}

}

}

}

Завдання 8\_2

using System;

using System.IO;

using System.Linq;

class Program

{

static void Main(string[] args)

{

string filePath = "E:\\f.txt";

// Зчитування чисел з файлу

string[] lines = File.ReadAllLines(filePath);

if (lines.Length > 0)

{

double[] numbers = lines.Select(double.Parse).ToArray();

// 1) Найбільше значення компонент

double maxNumber = numbers.Max();

Console.WriteLine($"Найбільше значення компонент: {maxNumber}");

// 2) Найменше значення компонент з парними номерами

double minEvenNumber = numbers.Where((num, index) => index % 2 == 0).Min();

Console.WriteLine($"Найменше значення компонент з парними номерами: {minEvenNumber}");

// 3) Найбільше значення модулів компонент з непарними номерами

double maxAbsOddNumber = numbers.Where((num, index) => index % 2 != 0).Select(Math.Abs).Max();

Console.WriteLine($"Найбільше значення модулів компонент з непарними номерами: {maxAbsOddNumber}");

// 4) Сума найбільшого і найменшого значень компонент

double sumMinMax = numbers.Min() + numbers.Max();

Console.WriteLine($"Сума найбільшого і найменшого значень компонент: {sumMinMax}");

// 5) Різниця першої і останньої компонент файлу

double firstNumber = numbers.FirstOrDefault();

double lastNumber = numbers.LastOrDefault();

double difference = lastNumber - firstNumber;

Console.WriteLine($"Різниця першої і останньої компонент файлу: {difference}");

}

else

{

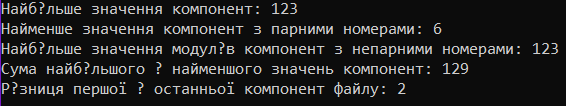
Console.WriteLine("Файл порожній або не існує.");

}

}

}

5. Результат роботи програми



6. [xanax4rl/lab (github.com)](https://github.com/xanax4rl/lab)